Paper No. 28

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

MAILED

SEP 3 0 2004

U.S. PATENT AND TRADEMARK OFFICE
BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte PAUL WESCHLER

Appeal No. 2003-1986
Application 09/315,200[1]

ON BRIEF

Before BARRETT, DIXON, and BARRY, Administrative Patent Judges.

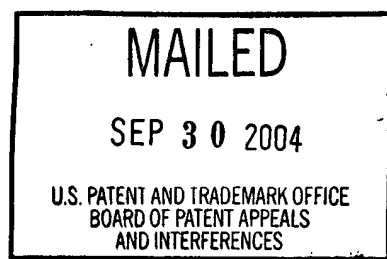BARRETT, Administrative Patent Judge.

DECISION ON APPEAL

This is a decision on appeal under 35 U.S.C. § 134(a) from

the final rejection of claims 1-53.

We reverse, but enter a new ground of rejection as to

claims 42-47.

---

[1] Application for patent filed May 19, 1999, entitled
"Mechanism and Method for Managing Service-Specified Data in a
Profile Service."

- 1 -

BACKGROUND

The invention relates to a profiling service for accessing user data using a plurality of profile objects.

Claim 1 is reproduced below.

1. A method for managing a profile service, the method comprising:

storing at least one true-data attribute in a profile object, said true-data attribute includes a true-data key and at least one true-data value field;

associating at least one meta-data attribute with said true-data attribute, said meta-data attribute includes a meta-data key and at least one meta-data value field, wherein the meta-data value field describes the associated true-data attribute;

storing said associated meta-data attribute in said profile object; and

managing said true-data attribute according to said associated meta-data attribute.

The examiner relies on the following reference:

Thomas   5,838,970   November 17, 1998

Claims 1-53 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Thomas.

We refer to the final rejection (Paper No. 14) (pages referred to as "FR__") and the examiner's answer (Paper No. 24) (pages referred to as "EA__") for a statement of the examiner's rejection, and to the amended appeal brief (Paper No. 23) (pages referred to as "Br__") and reply brief (Paper No. 25) (pages

referred to as "RBr___") for a statement of appellant's arguments thereagainst.

OPINION

Claims 1-53 are grouped to stand or fall together. Claim 1 is analyzed as representative.

The examiner's final rejection reads the limitations of claim 1 generally on column 3, lines 20-30, column 15, lines 5-35 and Table 4, and column 34, Table 7 (FR2-3), but does not specifically point to elements in Thomas that correspond to the claimed "true-data attribute" and "meta-data attribute" associated with the true-data attribute. The examiner's position is best explained in the response to the arguments section of the answer. The examiner finds that Table 4 of Thomas discloses a profile object that contains true-data attributes and meta-data attributes associated with the true-data attributes where the meta-data attributes can be used to manage the true-data attributes (EA11). The examiner finds that "Object Type #1" corresponds to a true-data attribute and the indented search parameters, such as "Object Type Name," "Object Type Version," "Last Update Time," etc., correspond to meta-data attributes associated with "Object Type #1" (EA11). The examiner notes that the "Last Update Time" in Table 4 of Thomas corresponds to the meta-data attribute "type=upd_98409245" for last update timestamp in appellant's Fig. 5A (see also specification at page 22) and,

therefore, it must be a meta-data attribute (EA12). The examiner

finds that attributes, such as "Object Type

Name=ACME::VIDEOMAIL," "Object Type Version," "Compression

Type=MPEG," and "Resolution=300 dpi" in Table 7, correspond to

meta-data attributes associated with "Object Type #1" and are

used to manage "Object Type #1" (EA13).

Appellant anticipated and addressed the examiner's position

in the brief. Appellant argues that Thomas fails to show a

method that: (1) associates at least one meta-data attribute with

a true-data attribute; (2) stores the associated meta-data

attribute in the profile object; and/or (3) manages the true-data

attribute according to the associated meta-data attribute (Br5).

It is argued that "[o]bject-type search parameters, the alleged

meta-data, are not used to manage other data in Table 4, but

instead are used to manage 'actual executables and libraries'

outside the Table 4" (Br7) and "there is no evidence that

meta-data attributes in a table are being used to manage a

true-data attribute in that same table" (emphasis omitted) (Br7).

It is argued that "if we are to characterize the 'search

parameters' and 'executable information' as meta-data, that

meta-data is not used to manage anything in the repository

itself, but instead to manage some other object that may or may

not even be executed on the same computer" (Br7). Appellant

argues that "[w]hat is missing from Thomas is any indication of

meta-data that would describe something about the true-data contents of each entry" (Br8). In the reply brief, appellant notes the examiner's characterization of specific attributes of Table 4 ("Object Type Name," "Object Type Version," "Last Update Time," etc.) as meta-data that are used to manage a true-data attribute, but argues that "[e]ach of these attributes specify some property of an external object, not properties that are used to manage the true data attribute itself" (footnote omitted) (RBr1). It is argued that the "Object Type Version" in Thomas refers to a version number associated with the external object, not a version number of the "Object Type #1" attribute (RBr2). It is argued that the "Last Update Time" attribute indicates when the external object is updated, not when the information repository is updated (RBr2). In summary, it is argued that "Thomas describes a system in which the attributes contained in an information repository describe external entities, not entities within the data structure itself" (RBr2).

We can understand why the examiner made and maintained the rejection. The tables in Thomas show data structures having parameters indented under headings and, as shown in Table 7, the headings and parameters are attributes because they are key/value pairs; thus, the tables closely resemble appellant's Fig. 5A in appearance. The examiner notes that the "Last Update Time" in Table 4 of Thomas is similar to the meta-data attribute

"type=upd_98409245" for last update timestamp in appellant's

Fig. 5A (see also specification at page 22) and concludes that it

must be a meta-data attribute (EA12). In addition, the examiner

was faced with the not so simple job of determining what is true

data and what is meta-data. As acknowledged by appellant, "[t]he

distinction between meta-data and true-data may be conceptually

difficult" (Br7). However, on balance, appellant has convinced

us of the correctness of his position.

Thomas does not use the terms "true data" and "meta-data,"

which makes our job difficult. Part of the problem is the

definition of terms. The specification defines an "attribute" as

a "key=value pair" (page 10). The specification defines

"true-data attribute" and "meta-data attribute" as (page 20):

> For clarity, a true-data attribute is defined herein as an
> attribute that contains a value (e.g. data, external
> reference, or binding) used by a user entity or client. A
> meta-data attribute is defined as an attribute associated
> with a true-data attribute which contains information used
> and maintained by the core profile engine 201 (shown in
> Fig. 2).

"True data" is the actual data to be used. "Meta-data" is data

about the "true data." The following definition of "meta-data"

is from "www.techweb.com/encyclopedia/":

> Data that describes other data. The term may refer to
> detailed compilations such as data dictionaries and
> repositories that provide a substantial amount of
> information about each data element. It may also refer to
> any descriptive item about data, such as the content of an
> HTML meta tag or a title field in a media file.

> Meta-data has existed for centuries. Card catalogs and
> handwritten indexes are examples long before the electronic
> age.
>
> Although countless articles, magazines and books spell this
> term as one word, "meta-data" with the hyphen is the proper,
> generic spelling, because the Metadata company has
> trademarked the name. See metadata, data dictionary,
> repository, meta tag and Meta Data Coalition.

"Meta-data" is also defined in "dictionary.reference.com/

search?q=meta-data":

> Data about data. In data processing, meta-data is
> definitional data that provides information about or
> documentation of other data managed within an application or
> environment.
>
> For example, meta-data would document data about data
> elements or attributes, (name, size, data type, etc) and
> data about records or data structures (length, fields,
> columns, etc) and data about data (where it is located, how
> it is associated, ownership, etc.). Meta-data may include
> descriptive information about the context, quality and
> condition, or characteristics of the data.

Traditionally, the information we put on the page is content or

true data. Meta-data is information used to describe or define

that content. So, it's not content in and of itself.

The examiner relies on both the implementation repository

data structure in Table 4 (probably because it contains a "Last

Update Time" parameter similar to appellant's "upd_" meta-data)

and on the service profile repository data structure in Table 7

(probably because the heading ."Object Type #1=ACME::VIDEOMAIL"

and the parameters like "Object Type Version=1" have the

appearance of attribute key/value pairs). We first examine the

implementation repository data structure (col. 14, line 5 to

col. 16, line 21 & Table 4; col. 35, line 33 to col. 36, line 33

& Table 9). As shown in Table 9, the search parameters can be

attributes with key/value pairs. The heading "Object Type #1"

does not have a value like "Object Type #1=ACME::VIDEOMAIL" in

Table 7, but we will assume that it does. We interpret

"ACME::VIDEOMAIL" to identify the object VIDEOMAIL within the

class ACME,[2] i.e., it identifies an external object type stored

somewhere else. Assuming "ACME::VIDEOMAIL" represents true data,

the object type entries contain the implementation information

needed to activate an object of that particular type and are not

meta-data attributes "wherein the meta-data value field describes

the associated true-data attribute," as claimed, because they do

not describe the data "ACME::VIDEOMAIL." When the implementation

repository receives a request from an object broker for

implementation information for an object of a particular type,

the search parameters are used to select an implementation

appropriate for the type of hardware, operating system, and other

implementation preferences required for the particular object

operation, and the information passed back from the

implementation repository includes that required to access the

---

[2] In object-oriented programming languages like C++, the
"::" operator is called the "binary scope resolution operator."
The class name and a double colon (e.g., Student::) are prepended
to each of the functions names. This prefix is a tag that tells
the compiler that you are referring to something inside the
Student class.

code to be executed and the libraries to be used in activating

the object implementation (col. 14, lines 16-27). Thus, the

search parameters select an appropriate implementation and the

executable information tells how to access the code for the

object implementation The search parameters and executable

information are not meta-data attributes "wherein the meta-data

value field describes the associated true-data attribute" because

they do not describe something about the term "ACME::VIDEOMAIL."

We also agree with appellant that the search parameters are

associated with an external object and are not used for "managing

said true-data attribute according to said associated meta-data

attribute." The same observations are made about the search

parameters in Table 7. Accordingly, we find that Tables 4, 7,

and 9 of Thomas do not anticipate claim 1. Since the claims

stand or fall together, the rejection of claims 1-53 is reversed.

New ground of rejection pursuant to 36 CFR § 41.50(b)

Claims 42-47 are rejected under 35 U.S.C. § 101 as being

directed to nonstatutory subject matter. The claims are directed

to a "profile object." The specification states (page 17,

lines 30-35): "As used herein, the term 'object' refers to a data

structure stored in mass storage or memory accessible by a

computer that contains specified data and a set of methods or

operations that enable the object to perform operations on the

data it contains." Therefore, an object is a data structure

- 9 -

containing data and methods for performing operations on the
data. Although the specification states that the object is
stored in mass storage or a computer memory, this is not an
express limitation of the claims. Implied limitations are not to
be read into the claims. An "object" per se has no physical
substance and does not fit within any of the statutory classes of
"process, machine, manufacture, or composition of matter" of
§ 101 because it does not recite steps or any physical structure.
See In re Warmerdam, 33 F.3d 1354, 1361-62, 31 USPQ2d 1754, 1760
(Fed. Cir. 1994) (data structure per se of claim 6 is not in one
of the categories of § 101). Furthermore, an "object" per se is
an "abstract idea" because it has no concrete physical
instantiation. We hold that the subject matter of claims 42-47
is nonstatutory because it does not fit within any of the
statutory categories and because it falls within the abstract
idea exception. If the claims were amended to recite that the
object was stored in a physical medium, this would overcome the
rejection because of the statutory nature of the medium and the
fact that the object information is functional when used in a
computer. See In re Lowry, 32 F.3d 1579, 32 USPQ2d 1031 (Fed.
Cir. 1994) (memory containing a stored data structure was
statutory subject matter); Manual of Patent Examining Procedure
§ 2106 IV.B.1 (distinguishing between "functional descriptive
material" and "nonfunctional descriptive material").

CONCLUSION

The rejection of claims 1-53 is reversed.

A new ground of rejection of claims 42-47 under 35 U.S.C. § 101 is entered pursuant to 37 CFR § 41.50(b).

This decision contains a new ground of rejection pursuant to 37 CFR § 41.50(b) (effective September 13, 2004, 69 Fed. Reg. 49960 (August 12, 2004), 1286 Off. Gaz. Pat. Office 21 (September 7, 2004)). 37 CFR § 41.50(b) provides "[a] new ground of rejection pursuant to this paragraph shall not be considered final for judicial review."

37 CFR § 41.50(b) also provides that the appellant, WITHIN TWO MONTHS FROM THE DATE OF THE DECISION, must exercise one of the following two options with respect to the new ground of rejection to avoid termination of the appeal as to the rejected claims:

> (1) Reopen prosecution. Submit an appropriate amendment of the claims so rejected or new evidence relating to the claims so rejected, or both, and have the matter reconsidered by the examiner, in which event the proceeding will be remanded to the examiner. . . .

> (2) Request rehearing. Request that the proceeding be reheard under § 41.52 by the Board upon the same record. . . .

No time period for taking any subsequent action in connection with this appeal may be extended under 37 CFR § 1.136(a).

<u>REVERSED - 37 CFR § 41.50(b)</u>

LEE E. BARRETT
Administrative Patent Judge)
)
)
)
)
) BOARD OF PATENT
JOSEPH L. DIXON) APPEALS
Administrative Patent Judge) AND
) INTERFERENCES
)
)
)
LANCE LEONARD BARRY)
Administrative Patent Judge)

HOGAN & HARTSON LLP
ONE TABOR CENTER, SUITE 1500
1200 SEVENTEEN ST.
DENVER, CO   80202